IN THE CLAIMS

1. (Currently Amended) A method for managing a code sequence, comprising:

determining first intermediate correlation values for a first plurality of sample sequences to load in a plurality of sample sequence registers during a first clock cycle;

determining second intermediate correlation values for the first plurality of sample sequences during a second clock cycle;

determining correlation outputs for the first plurality of sample sequences from the first and second intermediate correlation values; and

determining a synchronization point that identifies an amount of delay incurred from transmission of the sample sequences from the correlation outputs.

2. (Original) The method of Claim 1, wherein determining the first intermediate correlation values comprises processing coefficients in a first code sequence group in parallel with corresponding sample values in corresponding sample sequence groups from the first plurality of sample sequences.

3. (Original) The method of Claim 1, wherein determining the second intermediate correlation values comprises processing coefficients in a second code sequence group in parallel with corresponding sample values in corresponding sample sequence groups from the first plurality of sample sequences.

4. (Original) The method of Claim 1, wherein determining correlation outputs for the first plurality of sample sequences comprises taking a sum of the first and second intermediate correlation values for each of the first plurality of sample sequences.

5. (Original) The method of Claim 1, further comprising:

determining first intermediate correlation values for a second plurality of sample values during a third clock;

determining second intermediate correlation values for the second plurality of sample values during a fourth clock; and

determining correlation output values for the second plurality of sample value from the first and second intermediate correlation values.

6. (Currently Amended) A method for managing a code sequence, comprising:

processing a first group of coefficients in the code sequence, loaded in a plurality of code sequence registers during a first clock cycle, with a first group of contiguous sample values, loaded in a plurality of sample sequence registers, in a received sample to determine a first intermediate correlation value during the first clock cycle;

processing a second group of coefficients in the code sequence, loaded in the plurality of code sequence registers previously used for the first group of coefficients during a second clock cycle, with a second group of contiguous sample values, loaded in the plurality of sample sequence registers, in the received sample to determine a second intermediate correlation value during the second clock cycle;

determining a correlation output from the first and second intermediate correlation values; and

determining a synchronization point that identifies an amount of delay incurred from transmission of the sample values from the correlation output.

7. (Previously Presented) The method of claim 6, wherein the code sequence comprises L coefficients and the first and second group of coefficients in the code sequence each comprises n coefficients, where L and n may be any value.

8. (Original) The method of claim 7, wherein the first and second group of sample values in the received sample each comprises n sample values.

9. (Original) The method of claim 6, wherein the first and second group of coefficients in the code sequence are contiguous.

10. (Canceled)

11. (Original) The method of claim 6, wherein processing the first group of coefficient in the code sequence with the first group of sample values in the received sample comprises determining a sum of the products of the first group of coefficients in the code sequence with the first group of sample values in the received sample.

12. (Original) The method of claim 6, wherein processing the second group of coefficients in the code sequence with the second group of sample values in the received sample

comprises determining a sum of the products of the second group of coefficients in the code sequence with the second group of sample values in the received sample.

13. (Original) The method of claim 6, wherein determining the correlation output from the first and second intermediate correlation values comprises taking the sum of the first and second intermediate correlation values.

14. (Currently Amended) A method for managing a code sequence, comprising:

organizing the code sequence, having L contiguous coefficients, into L/n contiguous code sequence groups having n coefficients each, wherein n and L are ~~is~~ greater than 1;

selecting a number of sample sequences to process in parallel where each of the sample sequences has contiguous sample values from a received sample;

organizing contiguous sample values from each of a first set of contiguous sample sequences to process in parallel into a first set of contiguous sample sequence groups;

processing coefficients in each of the code sequence groups loaded into a plurality of code sequence registers in parallel with corresponding sample values in corresponding sample sequence groups from the first set of sample sequences loaded into a plurality of sample sequence registers, where each of code sequence groups is processed during a different clock cycle;

determining a correlation output for each of the sample sequences; and

determining a synchronization point that identifies an amount of delay incurred from transmission of the sample sequences from the correlation output.

15. (Original) The method of Claim 14, further comprising:

organizing contiguous sample values from each of a second set of sample sequences to process in parallel into a second set of contiguous sample sequence groups; and

processing coefficients in each of the code sequence groups in parallel with corresponding sample values in corresponding sample sequence groups from the second set of sample sequences, where each of the code sequence groups is processed during a different clock cycle.

16. (Canceled)

17. (Previously Presented) The method of Claim 16, wherein determining a synchronization point comprises determining a correlation output having a highest numerical value.

18. (Previously Presented) The method of Claim 14, wherein a first sample value in a first sample sequence includes a first sample value in the received sample and each consecutive sample sequence includes a next contiguous sample value in the received sample as a first sample value of the consecutive sample sequence.

19. (Original) The method of Claim 14, wherein processing comprises determining a sum of the products of the coefficients in each of the code sequence groups with each of the sample values in corresponding sample sequence groups from the first set of sample sequences.

20. (Original) The method of Claim 14, wherein the code sequence comprises a plurality of L contiguous values.

21. (Original) The method of Claim 20, wherein the code sequence is organized into a plurality of n code sequence groups.

22. (Original) The method of Claim 21, wherein a number, d, sample sequences are selected to process in parallel where each of the sample sequences has L contiguous sample values from the sample.

23. (Original) The method of Claim 22, wherein the first set of sample sequences is organized into a plurality of contiguous sample sequence groups having n values each.

24. (Canceled)

25. (Previously Presented) The method of Claim 14, wherein processing coefficients comprises processing coefficients for L/n clocks.

26. (Currently Amended) A method for managing a code sequence, comprising:

organizing the code sequence, having L contiguous coefficients, into L/n contiguous code sequence groups having n coefficients each, wherein n and L are ~~is~~ greater than 1;

selecting a number of sample sequences, d, to process in parallel where each of the sample sequences has L contiguous sample values from a received sample, where a first sample value in a first sample sequence is a first sample value in the received sample and each consecutive sample sequence includes a next contiguous sample value in the received sample as a first sample value in the consecutive sample sequence;

organizing sample values from each of a first set of d sample sequences into a first set of sample sequence groups having n values each and loading the set into a plurality of sample sequence registers;

processing coefficients in each of the code sequence groups loaded into a plurality of code sequence registers in parallel with corresponding sample values in corresponding sample sequence groups from the first set of d sample sequences, where each of the code sequence groups is processed during a different clock cycle, where ~~L and~~ d may be any value;

determining a correlation output for each of the sample sequences; and

determining a synchronization point that identifies an amount of delay incurred from transmission of the sample sequences from the correlation output.

27. (Previously Presented) The method of Claim 26, further comprising:

organizing sample values from each of a second set of d sample sequences into a second set of contiguous sample sequence groups having n values each; and processing values in each of the code sequence groups in parallel with corresponding sample values in corresponding sample sequence groups from the second set of d sample sequences, where each of the code sequence groups is processed during a different clock cycle.

28. (Canceled)

29. (Original) The method of Claim 28, wherein determining a synchronization output comprises determining a correlation output having a highest numerical value.

30. (Original) The method of Claim 26, wherein the code sequence is organized into L/n groups.

31. (Original) The method of Claim 26, wherein processing comprises determining a sum of the products of the coefficients in each of the code sequence groups with each of the sample values in corresponding sample sequence groups from the first set of d sample sequences.

32. (Original) The method of Claim 26, wherein the processing is completed after L/n clocks.

33. (Currently Amended) A correlator unit, comprising:

a plurality of code sequence registers that store coefficients from a code sequence group having n coefficients, the plurality of code sequence registers storing coefficients from one code sequence group of L/n code sequence groups at a time, where L is the number of coefficients in a code sequence, wherein n and L are~~is~~ greater than 1;

a plurality of sample registers that store sample values from a plurality of sample sequences that are processed in parallel;

a processing unit that processes coefficients in each of the plurality of code sequence groups in the plurality of code sequence registers in parallel with corresponding sample values in corresponding sample sequence groups from a first plurality of sample sequences in the plurality of sample registers, where each of the code sequence groups is processed to generate intermediate correlation values during a different clock cycle;

an accumulation unit that generates a correlation output for each of the sample sequences from the intermediate correlation values generated during the different clock cycles; and

a correlation output processor that determines a synchronization point that identifies an amount of delay incurred from transmission of the sample sequences from the correlation output.

34. (Previously Presented) The correlator unit of Claim 33, wherein the accumulation unit further comprises a plurality of accumulation sub-units each accumulation sub-unit receiving results from the processing unit for a designated sample sequence, each accumulation unit generating a correlation value for the designated sample sequence after each of the code sequence groups are processed.

35. (Previously Presented) The correlator unit of Claim 33, wherein the processing unit processes the coefficients in each of the plurality of code sequence groups in the plurality of code sequence registers in parallel with corresponding sample values in corresponding sample sequence groups from a second plurality of sample sequences in the plurality of sample registers, where each of the code sequence groups is processed during a different clock cycle.

36. (Canceled)

37. (Original) The correlator unit of Claim 36, wherein the correlation output processor determines a synchronization point from a correlation output having a highest numerical value.

38. (Previously Presented) The correlator unit of Claim 33, wherein the processing unit determines a sum of products of the coefficients in each of the code sequence groups with corresponding sample values in corresponding sample sequence groups.

39. (Currently Amended) A correlator unit, comprising:

a plurality of n code sequence registers that store n coefficients from a code sequence group, the plurality of n code sequence registers storing coefficients from one code sequence group of L/n code sequence groups at a time, where L is the number of coefficients in a code sequence, wherein n and L are~~is~~ greater than 1;

a plurality of n+d-1 sample registers that store sample values from a plurality of d sample sequences that are processed in parallel, wherein d is greater than 1; and

a processing unit that processes coefficients in each of the plurality of code sequence groups in the plurality of n code sequence registers in parallel with corresponding sample values in corresponding sample sequence groups from a first plurality of d sample sequences in the plurality of n+d-1 sample registers, where each of the code sequence groups is processed to generate intermediate correlation values during a different clock cycle, wherein d may be any value;

an accumulation unit that determines a correlation output for each of the sample sequences from the intermediate correlation values generated during the different clock cycles; and

a correlation output processor to determine a synchronization point that identifies an amount of delay incurred from transmission of the sample sequences from the correlation output.

40. (Previously Presented) The correlator unit of Claim 39, wherein the accumulation unit further comprises an accumulation sub-unit, corresponding to each of the d sample sequences that are processed in parallel, that receives results from the processing unit for a designated sample sequence and that determines a correlation output for the designated sample sequence after each of the code sequence groups are processed.

41. (Original) The correlator unit of Claim 39, wherein the processing unit processes the coefficients in each of the plurality code sequence groups in the plurality of n code sequence

registers in parallel with corresponding sample values in corresponding sample sequence groups from a second plurality of d sample sequences in the plurality of n+d-1 sample registers, where each of the code sequence groups is processed during a different clock cycle.

42. (Original) The correlator unit of Claim 40, further comprising correlation output processor that determines a synchronization point for the code sequence from the correlation outputs.

43. (Original) The correlator unit of Claim 42, wherein the correlation output processor determines a synchronization point from a correlation output having a highest numerical value.

44. (Original) The correlator unit of Claim 39, wherein the processing unit determines a sum of products of the coefficients in each of the code sequence groups with each of the sample values in corresponding sample sequence groups from the first set of d correlation sequences.

45. (Original) The correlator unit of Claim 39, wherein the processing is completed after L/n clocks.

46. (Currently Amended) A correlator unit, comprising:

means for storing coefficients from a code sequence group having n coefficients, the means for storing coefficients from one code sequence group of L/n code sequence groups at a time, where L is a number of coefficients in a code sequence, wherein n and L areis greater than 1;

means for storing sample values from a plurality of sample sequences that are processed in parallel;

means for processing coefficients in each of the plurality of code sequence groups in the means for storing coefficients in parallel with corresponding sample values in corresponding sample sequence groups from a first plurality of contiguous sample sequences in the means for storing sample values, where each of the code sequence groups is processed to generate intermediate correlation values during a different clock cycle;

means for determining a correlation output for each of the sample sequences from the intermediate correlation values generated during the different clock cycles; and

means for determining a synchronization point that identifies an amount of delay incurred from transmission of the sample sequences from the correlation output.

47. (Previously Presented) The method of Claim 1, wherein determining the synchronization point comprises identifying a correlation output having a highest numerical value.

48. (Previously Presented) The method of Claim 6, wherein the first group of contiguous sample values are loaded into a set of sample sequence registers during the first clock cycle and the second group of contiguous sample values are loaded into the set of sample sequence registers during the second clock cycle.

49. (Previously Presented) The method of Claim 6, wherein determining the synchronization point comprises identifying a correlation output having a highest numerical value.

50. (Previously Presented) The method of Claim 14, wherein each of the code sequence groups are loaded into the same set of code sequence registers.